MANUALE OPERATIVO E DIZIONARIO BASIC

LASER 310
COLOR COMPUTER

SCUOLA
Scheidegger

MANUALE OPERATIVO E DIZIONARIO BASIC

Questo volume contiene le informazioni di carattere generale relative alla installazione del computer, ed ad alcune convenzioni usate nel linguaggio di programmazione BASIC.

Nella seconda parte viene esposto in ordine alfabetico il linguaggio BASIC relativo ai COMPUTER LASER 110/210/310/500.

Poichè la maggior parte delle istruzioni si riferisce al BASIC standard questo manuale è di grande utilità a chiunque possieda un computer.

SOMMARIO

GENERALITA'

DIZIONARIO BASIC

DIZIONARIO BASIC (segue)

INTRODUZIONE

Questo manuale contiene le informazioni di base per poter operare con i computer LASER della serie 110/210/310/500.

Si consiglia una attenta lettura di questa introduzione, per comprendere le parti seguenti del testo.

Ogni computer e' dotato di un manuale di uso, che spesso non e' scritto in lingua italiana. E' molto importante fare riferimento al manuale originale, anche se e' in altra lingua, per consultare tabelle, e schemi.

Le informazioni sono da considerare valide per tutti i computers LASER, salvo diversa indicazione.

Le caratteristiche NON comuni a tutti i modelli LASER sono segnalate con l'indicazione del modello che possiede, o non possiede tali caratteristiche.

Esempi: (LASER 500) segnala che solo il LASER 500 ha quella caratteristica.

(ESCLUSO LASER 110/210) segnala che le informazioni non riguardano i modelli 110 e 210

Per quanto riguarda il dizionario BASIC le numerose voci sono elencate in ordine alfabetico, per facilitare la ricerca.

Per ogni voce viene indicato:

Tipo: da' la classificazione della voce (istruzione, comando , funzione etc.)

Ortografia: indica la corretta scrittura del comando, precisando i parametri occorrenti in forma generica. E' possibile trovare indicazioni come : [lista] oppure (x$,i,j) etc.

Per maggiore chiarezza i parametri che devono essere precisati dal programmatore, o che possono assumere forme diverse sono indicati con lettere minuscole o parole in minuscolo racchiuse tra parentesi quadrate.

Le parentesi quadrate NON fanno parte del comando o istruzione, se invece ci sono parentesi rotonde queste devono essere trascritte in quanto parte dell'istruzione.

Esempi:

[nome variabile] indica che si deve mettere il nome di una variabile. Le parentesi quadrate NON devono essere trascritte.

DIM x$(n), .. , .. in questo caso x$ ed n sono parametri che devono essere precisati dall'utente, ed i puntini indicano che e' possibile ripetere i parametri separandoli con una virgola.

Note: precisano i limiti e le caratteristiche dei parametri indicati in "Ortografia".

Uso: precisa le modalita' d'uso.

Esempio: per ogni voce viene indicato uno o piu' esempi di facile comprensione.

Siamo sicuri che potrete trovare un valido aiuto in questo manuale; anche se in un primo tempo troverete qualche difficolta' nell'uso del vostro computer, e del registratore a cassette cercate di applicarvi con maggiore impegno, per superare i piccoli inconvenienti dovuti alla vostra inesperienza. In breve tempo eliminerete ogni problema.

E' inteso che occorre applicarsi molto, con buona volonta' e spirito intraprendente, eseguendo prove e verifiche dei vari comandi, ricordando che e' molto piu' probabile un vostro errore che un cattivo funzionamento del computer.

Se qualcosa non va nel modo previsto verificate il vostro operato, prendete nota di ogni particolare del vostro comportamento, e verificate su questo manuale e sul manuale allegato al computer se avete eseguito le giuste operazioni.

Solo dopo NUMEROSE e DOCUMENTATE prove di insuccesso e' possibile attribuire la colpa alle apparecchiature, che nella maggior parte dei casi non hanno alcun inconveniente.

Vi segnaliamo questo perche' TUTTE le persone che hanno il primo incontro con i computers tendono ad attribuire le loro colpe al computer, piuttosto che alla propria inesperienza.

A questo proposito vi saremo grati se vorrete segnalarci tempestivamente qualsiasi inesattezza o imprecisione che troverete in questo manuale, anche se da parte nostra abbiamo verificato e controllato molte volte il testo.

GENERALITA'

L'INSTALLAZIONE DEL COMPUTER

Un HOME-COMPUTER e' costituito da un insieme di apparecchiature che devono essere collegate tra loro, e che ricevono alimentazione dalla rete elettrica a 220 Volt.
Le principali componenti sono:
- tastiera/CPU
- video (MONITOR o TV)
- registratore a cassette o drive per floppy disk
- stampante o plotter
Le apparecchiature di base sono la tastiera/CPU, ed il video.

Per installare il Vostro computer seguite gli schemi di collegamento riportati nel manuale del computer, ricordando quanto segue:

la tastiera e' gia' collegata al computer, poiche' e' inserita nello stesso contenitore della scheda che contiene i microprocessori (CPU).

Per poter collegare la tensione di rete 220 V usare l'alimentatore apposito.

Il registratore deve essere collegato con un proprio alimentatore alla rete elettrica, e con uno speciale cavetto al computer. Le entrate ed uscite per il registratore sono solitamente contrassegnate con scritte come: TAPE oppure REC.

Il computer deve essere collegato con apposito cavetto coassiale alla presa di antenna di un televisore B/N o colore.
In relazione al tipo di TV usato commutare il deviatore posto sotto alla tastiera con l'indicazione B/W (BIANCO e NERO), oppure su C (COLORE).

Il cavo che collega il computer al televisore deve essere collegato alla giusta uscita del computer, contraddistinta dalla scritta TV.
Se si usa un MONITOR usare invece l'uscita con la scritta MONITOR.

Il televisore deve essere collegato alla rete elettrica con il proprio cavo, inoltre e' necessario SINTONIZZARSI sul segnale emesso dal computer, ricercandolo come se fosse una stazione televisiva, (QUANDO IL COMPUTER E' ACCESO E COLLEGATO ALLA TV)

Se avete difficolta' nel trovare il segnale o a memorizzarlo sul Vs. televisore rivolgetevi al Vs. tecnico TV.

Quando tutto e' correttamente collegato, sul televisore compare un riquadro, ed alcune scritte relative al computer usato, inoltre compare un quadratino lampeggiante (chiamato CURSORE) e la scritta READY. che indica che il computer e' pronto per ricevere comandi.

I COMANDI IN MODO DIRETTO

I computer LASER possono operare anche nel modo DIRETTO, per cui e' possibile dare la maggior parte dei comandi scrivendo il comando e premendo il TASTO IMMISSIONE (ENTER o RETURN). Il comando verra' immediatamente eseguito.

ESEMPIO: PRINT 10 (piu' tasto ENTER o RETURN) provochera' la scrittura sul video del numero 10.

Prima di poter operare occorre pero' conoscere i diversi comandi.

Se digitate qualcosa di errato e premete RETURN il computer segnala ? SYNTAX ERROR

Non preoccupatevi: non e' possibile danneggiare il computer premendo i tasti. Se inviate comandi errati il computer segnalera' di non comprenderli, ma non subira' alcun danno.

Controllate quali sono i comandi che il Vs. computer e' in grado di riconoscere, e soprattutto state attenti a scriverli nel modo esatto.

I TASTI DI SELEZIONE DI CONTROLLO E DI EDITING.

La tastiera dei computer LASER e' provvista di numerosi tasti che possono svolgere una o piu' funzioni.

Il tasto contrassegnato con la scritta SHIFT permette di SELEZIONARE il modo MAIUSCOLO, (LASER 500), oppure di scegliere i simboli grafici (LASER 110/210/310).

Il tasto CTRL permette di SELEZIONARE i comandi pre-programmati, che semplificano l'uso del linguaggio BASIC.

Il tasto SHIFT ed il tasto CTRL devono essere premuti ASSIEME al tasto prescelto.

ESEMPIO: CTRL P (tasto CTRL con tasto P) provoca la scrittura dell' istruzione PRINT sul video.

CTRL C (tasto CTRL con tasto C) nel LASER 500 equivale ai tasti CTRL con BREAK dei computer LASER 110/210/310 e provoca l'interruzione di un programma.

I tasti per i movimenti del CURSORE sono contraddistinti da frecce (destra sinistra, alto, basso).

Il LASER 500 ha tasti appositi, posti in basso a destra della tastiera. Per gli altri computer LASER si ha invece:

CTRL . (tasto CTRL con tasto .) muove in ALTO
CTRL , (tasto CTRL con tasto ,) muove a DESTRA
CTRL M (tasto CTRL con tasto M) muove a SINISTRA
CTRL SPACE (tasto CTRL con barra SPAZIO) muove in BASSO

Il LASER 500 ha anche altri tasti speciali:
INS permette di inserire un carattere in una parola
DEL permette di cancellare un carattere
DEL LINE cancella dalla posizione del cursore a fine riga
CLS HOME cancella lo schermo (se premuto con tasto SHIFT), oppure posiziona in alto a sinistra (HOME) il CURSORE (se premuto da solo).

Nei computer LASER 110/210/310 si hanno queste possibilita':
CTRL L (tasto CTRL con tasto L) permette di inserire un carattere in una parola.
CTRL ; (tasto CTRL con tasto ;) cancella un carattere
CTRL : (tasto CTRL con tasto :) abilita scritte REVERSE, oppure disabilita (se precedentemente inserito)
CTRL H (tasto CTRL con tasto H) cancella il video

Altri tasti speciali del LASER 500 sono:
BS (BACK SPACE) fa tornare indietro il cursore di un posto.

GRAPH (GRAPHIC) abilita i caratteri grafici predefiniti, vedere tabella dei caratteri grafici sul manuale.

CAP LOCK (CAPITALS LOCK) abilita e mantiene bloccato il MODO MAIUSCOLE (per disabilitare premere di nuovo).

RESET ripristina il sistema se premuto con SHIFT.

F1 F2 ... F10 tasti di FUNZIONE che possono essere usati anche con il tasto SHIFT ed il tasto CTRL (Vedere la funzione KEY nel dizionario BASIC).

ESC (tasto di ESCAPE) permette di definire le "sequenze di ESCAPE" per ottenere particolari funzioni, ad esempio premendo il tasto ESC con il tasto B si abilita il MODO REVERSE di scrittura sul video (caratteri chiari su fondo scuro).
Vedere il manuale del computer per le sequenze ammesse.

TAB (tabulazione) sposta il cursore al prossimo punto di tabulazione. Le colonne 9,17,25,33 (nel modo 40 colonne) e le colonne 9,17,25,33,41,49,57,65,73 (nel modo 80 colonne) sono i punti di tabulazione predefiniti.

Per impratichirsi nell'uso della tastiera seguire le esercitazioni e gli schemi riportati nel manuale del Vs. computer.

COSTANTI E VARIABILI

I computer LASER sono in grado di operare con il linguaggio BASIC, ed adottano le regole di tale linguaggio per il riconoscimento dei valori.

In particolare e' possibile operare con COSTANTI e VARIABILI di tipo diverso come segue:

COSTANTI NUMERICHE sono numeri e possono essere rappresentati nel modo voluto dall'utente. Le cifre decimali di un numero devono essere separate dalla parte intera con un puntino e non con la virgola.

Esempio di COSTANTI NUMERICHE: 10 12.55 3.141592654

COSTANTI ALFANUMERICHE (STRINGHE) sono l'insieme di uno o piu' caratteri o simboli o cifre, e devono essere SEMPRE racchiuse tra virgolette.

Esempio di COSTANTI ALFANUMERICHE: "BRAVO" "TEL. 031/27.55.88"

COSTANTI ESADECIMALI (LASER 500) contraddistinte dai simboli &H posti davanti ad un numero esadecimale

---

VARIABILI NUMERICHE

Le variabili ammesse dai computer LASER possono essere di tipo diverso, e sono segnalate da un NOME e da un SIMBOLO di riconoscimento del tipo.

NUMERICHE INTERE contraddistinte dal segno % (percento)
esempio: A% B1% COD%

NUMERICHE REALI in semplice precisione non hanno alcun sibmolo di riconoscimento esempio: A ZZ Z1 C1P

NUMERICHE REALI in DOPPIA PRECISIONE (LASER 500) contraddistinte dal segno # (numero) esempio: A# B1# C#

La differenza tra i diversi tipi e' la seguente:
le variabili INTERE non hanno alcuna cifra decimale, e sono rappresentate da un solo BYTE, per valori compresi tra -32768 e +32767

le variabili REALI in SINGOLA PRECISIONE possono avere una parte intera ed una parte decimale e sono rappresentate da un solo BYTE.

le variabili REALI in DOPPIA PRECISIONE possono avere una parte intera ed una decimale, e sono rappresentate in memoria da 2 BYTES.

VARIABILI ALFANUMERICHE

Sono identificate dal simbolo $ (DOLLARO) che segue il nome.
Esempio: A$ PI$ AB$

REGOLE PER L'ATTRIBUZIONE DEL NOME ALLE VARIABILI

Il NOME di una variabile di qualsiasi tipo DEVE iniziare con una lettera dell'alfabeto.

I computer LASER 110/210/310 possono identificare nomi che abbiano al massimo 2 caratteri.
Se il nome e' piu' lungo di 2 caratteri, il nome che il computer ritiene valido e' costituito solo dai primi due caratteri; esempio: PROVA$ e PRO$ equivalgono ad un unico nome PR$ che e' l'unico riconosciuto.

Il LASER 500 e' in grado di riconoscere qualsiasi nome, per qualunque lunghezza; esempio: IMPORTO# ed IMPORTI# sono due nomi diversi e riconosciuti.

Il NOME puo' contenere delle cifre, ma il primo carattere e' sempre una lettera dell'alfabeto; es: A1$ A2%

Il nome che NON e' seguito da alcun simbolo identifica VARIABILI NUMERICHE IN SEMPLICE PRECISIONE.
Se dopo il nome si mette un simbolo e' necessario che sia uno dei simboli accettati dal computer.

Il NOME di una VARIABILE DEVE SEMPRE CONCORDARE con il CONTENUTO della VARIABILE STESSA, altrimenti si verifica un errore.

Il computer LASER 500 ammette la pre-definizione del TIPO di variabili (vedere istruzione DEF INT/SGN/DBL/STR).

SEGNI DI SINTASSI

I computer LASER accettano il segno : (DUE PUNTI) come separatore delle istruzioni, per cui e' possibile dare piu' di una istruzione per ogni riga, separandole con due punti.

Il segno virgola (comma) ed il punto e virgola (semi-colon) sono interpretati come segni di interpunzione e devono essere usati nel modo richiesto dall' ORTOGRAFIA delle diverse istruzioni.
Con l'istruzione PRINT e derivate il punto e virgola annulla il ritorno a capo, mentre la virgola provoca la tabulazione.

FORMATO VIDEO

I computer LASER 110/210/310 determinano una configurazione video di 32 colonne x 16 righe in modo testo, oppure una configurazione 128 x 64 pixels in modo grafico.

Il LASER 500 ha la possibilita' di operare con diverse configurazioni di video: vedere funzioni GR e TEXT.

OPERATORI RELAZIONALI

Gli operatori relazionali riconosciuti dal BASIC dei computer LASER sono rappresentati con i seguenti simboli:

>	MAGGIORE	>=	MAGGIORE O UGUALE
<	MINORE	>=	MINORE O UGUALE
=	UGUALE	<>	DIVERSO

Per valori numerici il confronto e' fatto in relazione al valore algebrico dei numeri, per cui il numero -1 e' MINORE del numero 1 etc.

Per valori alfanumerici (stringhe) viene confrontato un carattere per volta, fino alla determinazione del risultato, in relazione al valore del CODICE ASCII dei caratteri, e non sulla LUNGHEZZA della parola.

Esempi:

"ALBO" e' MAGGIORE di "ALBERO" poiche' la lettera O di ALBO ha cod. ASCII piu' alto della lettere E di ALBERO.

"S.FRANCESCO" e' MINORE di "SAN FRANCESCO" poiche' il punto ha cod. ASCII 46 mentre la lettera A di SAN ha cod. ASCII 65.

OPERATORI LOGICI

Gli operatori logici riconosciuti sono i seguenti:

NOT (negazione) esempio: A NOT B

OR (esclusione) esempio: A OR B

AND (contemporaneita') esempio: A AND B

Il LASER 500 dispone anche dei seguenti operatori logici:

XOR (OR esclusivo) esempio: A XOR B

IMP (implicazione) esempio: A IMP B

EQV (equivalenza) esempio: A EQV B

Per verificare i risultati dei diversi operatori consultare le TAVOLE DI VERITA' (TRUTH TABLES) sul manuale del computer.

Esempio dell'uso degli operatori logici e relazionali:

```
10 INPUT "INSERIRE TRE NUMERI";A,B,C
20 IF A=B AND B=C THEN PRINT "A=B=C"
30 IF (NOT A=B) OR (NOT B=C) THEN PRINT "A=B=C E' FALSO"
40 END
RUN
INSERIRE TRE NUMERI  10,5,7
A=B=C E' FALSO
```

Solitamente gli operatori logici e gli operatori relazionali vengono usati con i test condizionali IF ... THEN ... ELSE oppure con il ciclo WHILE ... WEND .
Vedere queste voci nel dizionario BASIC.

OPERATORI MATEMATICI E PARENTESI

I simboli usati per identificare le operazioni matematiche sono i seguenti:

+ (piu')	SOMMA
- (meno)	SOTTRAZIONE
* (asterisco)	MOLTIPLICAZIONE
/ (barra diag.)	DIVISIONE
(freccia in alto)	ELEVAMENTO A POTENZA

Il simbolo = (uguale) e' usato come segno di ASSEGNAZIONE, e sottointende l' istruzione LET (vedere questa istruzione).

La priorita' delle operazioni e' quella usuale del linguaggio BASIC :

1- ATTRIBUZIONE DEL SEGNO
2- ELEVAMENTO A POTENZA
3- MOLTIPLICAZIONE E DIVISIONE
4- SOMMA E SOTTRAZIONE

A parita' di priorita' viene eseguita per prima l'operazione che si trova piu' a sinistra, in quanto le operazioni sono eseguite da sinistra verso destra.

Le PARENTESI ROTONDE () modificano la priorita' delle operazioni, per cui viene attribuita maggiore precedenza alle operazioni dentro le parentesi piu' interne.

Nel caso di piu' livelli di parentesi, vengono eseguite prima le operazioni dentro le parentesi piu' interne.

Esempio: PRINT 100/((5*4)/(2+3))-30/2   il risultato e' 10.

NUMERAZIONE BINARIA ED ESADECIMALE

1- Premessa: il sistema decimale

Il metodo comunemente usato per contare adotta il sistema decimale. Questo sistema si serve di simboli particolari, detti cifre, per rappresentare dei valori numerici.

Le cifre adottate dal sistema decimale sono le sèguenti:

0	ZERO	5	CINQUE
1	UNO	6	SEI
2	DUE	7	SETTE
3	TRE	8	OTTO
4	QUATTRO	9	NOVE

si tratta, come tutti sappiamo di dieci simboli, uno dei quali e' il simbolo 0 (ZERO) che identifica il nulla.
Il sistema si chiama DECIMALE (con base dieci), proprio perche' adotta dieci simboli per rappresentare i numeri. I simboli usati sono anche chiamati UNITA'.

Iniziando a contare dal primo simbolo (ZERO), e aggiungendo ogni volta un valore intero (UNO), si possono usare i dieci simboli per contare fino a 9 (NOVE), usando una dopo l'altra tutte le UNITA'.

Quando non si dispone di ulteriori unita', occorre applicare un semplice accorgimento, chiamato RIPORTO, che consiste nel riportare a ZERO le unita', affiancando alla cifra delle UNITA' una nuova cifra che indica il RIPORTO.
Dopo il numero 9 potremo continuare a contare, mettendo a 0 le UNITA', e portando il RIPORTO a sinistra delle unita'.

Otteniamo in questo modo il numero 10 (DIECI).
Trattandosi di sistema DECIMALE il posto occupato dal segno 1 viene detto casella delle DECINE, ed indica il numero di volte che si e' gia' contato da 0 a 9 (in questo caso una sola volta).

Proseguendo a contare avremo, come risaputo:

11 UNDICI
12 DODICI
13 TREDICI
14 QUATTORDICI
15 QUINDICI
16 SEDICI
17 DICIASSETTE
18 DICIOTTO
19 DICIANNOVE

dopo il 19 (DICIANNOVE) e' necessario eseguire ancora il RIPORTO, per cui la cifra delle UNITA' viene riportata a ZERO, ed una unita' viene aggiunta alla casella delle DECINE per indicare che si e' contato un' altra volta da 0 a 9.
Per rappresentare il nuovo numero metteremo quindi la cifra 2 (DUE) nella casella delle DECINE e la cifra 0 nella casella delle UNITA' ottenendo il numero 20 (VENTI).

Come tutti sanno e' possibile continuare all'infinito, semplicemente riportando ogni volta i valori in caselle adiacenti che per il sistema decimale prendono il nome di CENTINAIA, MIGLIAIA, DECINE DI MIGLIAIA, CENTINAIA DI MIGLIAIA, MILIONI, etc.

Il numero 567 indica 5 CENTINAIA piu' 6 DECINE piu' 7 UNITA'

In matematica le diverse caselle sono identificate dagli esponenti di 10, nel seguente modo:

10 elevato a 0 = 1 (per convenzione) = UNITA'
10 elevato a 1 = 10 = DECINE
10 elevato a 2 = 100 = CENTINAIA
.
.
10 elevato a 6 = 1000000 = MILIONI

e cosi' di seguito; si puo' cosi' notare che ogni posizione (casella) occupata dalle cifre e' in relazione ad un esponente di 10. Il valore 10 da cui deriva il nome del sistema e' detto anche BASE della NUMERAZIONE, ed indica semplicemente il NUMERO di SIMBOLI che si usano per rappresentare i valori, come gia' visto sopra.

2- Sistema binario

Il sistema binario adotta BASE 2 per rappresentare i numeri, e da quanto visto precedentemente e' intuibile che il sistema binario dispone di solo DUE simboli per rappresentare i numeri.

I simboli usati sono 0 (ZERO) e 1 (UNO).

Le cifre di un numero binario identificano quindi le potenze di due, nel seguente modo:

2 elevato a 0 = 1 (convenzione) CIFRA PIU' A DESTRA ...x
2 elevato a 1 = 2 SECONDA CIFRA DA DESTRA ..x.
2 elevato a 2 = 4 TERZA CIFRA DA DESTRA .x..

e cosi' di seguito; il valore di un numero binario e' dato dalla somma delle potenze di due.

Volendo conoscere il valore decimale del numero BINARIO 1011 si dovra' operare da destra verso sinistra, come segue:

1 volta 2 elevato a 0 = 1 x 1 = 1 piu'
1 volta 2 elevato a 1 = 1 x 2 = 2 piu'
0 volte 2 elevato a 2 = 0 x 4 = 0 piu'
1 volta 2 elevato a 3 = 1 x 8 = 8

totale 1 + 2 + 0 + 8 = 11 (valore decimale di 1011 binario)

Applicando lo stesso meccanismo usato per la numerazione decimale avremo la seguente numerazione binaria:

0	ZERO
1	UNO
10	UNO-ZERO
11	UNO- UNO
100	UNO-ZERO-ZERO
101	UNO-ZERO- UNO
110	UNO- UNO-ZERO
111	UNO- UNO- UNO
1000	UNO-ZERO-ZERO-ZERO

in questo caso e' evidente che il RIPORTO delle cifre avviene piu' frequentemente, poiche' si dispone di soli due simboli per rappresentare i numeri (0 e 1).

Confrontando il sistema decimale con il sistema binario avremo:

Esponente ------->	7	6	5	4	3	2	1	0
VALORI BASE 10 -->	10000000	1000000	100000	10000	1000	100	10	1
VALORI BASE 2 --->	128	64	32	16	8	4	2	1

DECIMALE	BINARIO	DECIMALE	BINARIO
0	0	8	1000
1	1	9	1001
2	10	10	1010
3	11	11	1011
4	100	12	1100
5	101	13	1101
6	110	14	1110
7	111	15	1111

Il sistema binario necessita di un maggior numero di CASELLE per identificare un numero, poiche' dispone solo delle cifre 0 e 1.

3- Sistema esadecimale

Il termine ESA-DECIMALE sta ad indicare la BASE 16, per cui questo sistema di numerazione adotta 16 simboli per rappresentare i numeri. I simboli usati sono:

0 1 2 3 4 5 6 7 8 9 A B C D E F

per la rappresentazione dei valori valgono le regole gia' viste sopra, per cui avremo questa numerazione:

```
0 ZERO           8 OTTO
1 UNO            9 NOVE
2 DUE            A A
3 TRE            B BI
4 QUATTRO        C CI
5 CINQUE         D DI
6 SEI            E E
7 SETTE          F EFFE
```

dopo EFFE si prosegue come gia' visto con il RIPORTO:

```
10 UNO ZERO
11 UNO  UNO
12 UNO  DUE
 .
 .
 .
1A UNO A
1B UNO BI
 .
 .
1F UNO EFFE
```

Ogni cifra identifica gli esponenti di 16 nel seguente modo:

```
esponente ---->   4      3     2    1    0
valore BASE 16  65536  4096   256  16    1
```

per cui il numero esadecimale 10AF (UNO ZERO A EFFE) rappresenta il seguente valore decimale:

```
F (cioe' 15) volte 16 elevato a 0 = 15 x    1 =   15 piu'
A (cioe' 10) volte 16 elevato a 1 = 10 x   16 =  160 piu'
0 (ZERO)     volte 16 elevato a 2 =  0 x  256 =    0 piu'
1            volta 16 elevato a 3 =  1 x 4096 = 4096
                                               ---------
                                  TOTALE:      4271
```

per cui 10AF equivale al valore 4271 decimale.

Nota: per identificare senza possibilita' di errore i numeri esadecimali si e' soliti scrivere tali numeri con una H (ACCA) maiuscola che segue il numero (es: 3256H C000H etc.).

In altri casi tali numeri sono identificati dal segno $ (DOLLARO) che precede il numero (es: $C1FA $000F etc.).

ABS

Tipo: FUNZIONE NUMERICA

Ortografia: ABS(x)

Note: x e' un argomento numerico.

Uso: Da' il valore ASSOLUTO dell'argomento numerico.

Esempio:

```
10 A = - 12.55 : B = 24.32
20 PRINT A;ABS(A)
30 PRINT B;ABS(B/2)
40 END
RUN
-12.55   12.55
 24.32   12.16
```

ASC

Tipo: FUNZIONE DI CONVERSIONE

Ortografia: ASC(x$)

Note: x$ e' un argomento alfanumerico. La funzione viene eseguita solo sul primo carattere di costanti o variabili alfanumeriche che non abbiano valore nullo.

Uso: Da' il codice ASCII del primo carattere dell'argomento alfanumerico. (Verificare sul manuale del computer i codici ASCII).

Esempio:

```
10 PRINT ASC("C")
20 Z$="ABC" : PRINT ASC(Z$)
RUN
 67
 65
```

ATN

Tipo: FUNZIONE TRIGONOMETRICA

Ortografia: ATN(x)

Note: x e' un valore numerico che indica l'angolo espresso in radianti.

Uso: Da' il valore dell'arcotangente dell'angolo.

ESEMPIO:

```
10 A=ATN(3.14/2) : PRINT A
RUN
1.00366
```

AUTO (LASER 500)

Tipo: COMANDO DI SISTEMA

Ortografia: AUTO n1,n2

Note: n1 indica il numero di riga di partenza
n2 indica l'intervallo di numerazione.

Uso: Consente la numerazione automatica delle righe di programma durante la programmazione.
Per interrompere premere il tasto CTRL piu' il tasto C.

Esempio: AUTO 100,10

BREAK

Tipo: COMANDO DIRETTO (usare gli appositi tasti)

Uso: LASER 110/210/310 : TASTO CTRL+TASTO BREAK
LASER 500 : TASTO CTRL + TASTO C

CALL (LASER 500)

Tipo: FUNZIONE DI SISTEMA

Ortografia: CALL nv (I,J,K)

Note: nv e' il nome di una variabile (che non puo' essere una variabile con indice), che contiene l'indirizzo di partenza di una subroutine.
Gli argomenti I,J,K sono valori che vengono "passati" alla subroutine, e devono essere contenuti in variabili.

Uso: Esegue un salto con ritorno alla subroutine di cui e' specificato l'indirizzo.

Esempio:

```
10 MYROUT = &HD000
20 CALL MYROUT (I,J,K)
```

CDBL (LASER 500)

Tipo: FUNZIONE DI CONVERSIONE

Ortografia: CDBL (x)

Note: x e' un argomento numerico.

Uso: Converte in doppia precisione un valore in singola precisione.

Esempio:
```
10 A = 454.67
20 PRINT A;CDBL(A)
RUN
454.67  454.670013427344
```

CHR$

Tipo: FUNZIONE DI CONVERSIONE

Ortografia: CHR$(x)

Note: x e' un argomento numerico intero compreso tra 0 e 255.

Uso: Da il carattere corrispondente al codice ASCII indicato. (Verificare sul manuale del computer i codici ASCII).
Questa funzione e' l'opposto della funzione ASC(x).

Esempio:
```
10 PRINT CHR$(65)
20 X=66  : PRINT CHR$(X)
RUN
A
B
```

CINT (LASER 500)

Tipo: FUNZIONE DI CONVERSIONE

Ortografia: CINT(x)

Note: x e' un valore o una espressione numerica compreso tra -32768 e +32767.

Uso: Converte in intero, con arrotondamento, un valore decimale.

Esempio:
```
10 A = 454.67
20 PRINT A;CINT(A)
RUN
454.67  455
```

CLEAR

Tipo:	COMANDO DI SISTEMA

LASER 110/210/310:

Ortografia:	CLEAR nn
Note:	nn e' un valore numerico.
Uso:	Imposta a zero le variabili numeriche, a valore nullo le variabili alfanumeriche, e riserva spazio in memoria per variabili alfanumeriche.
ESEMPIO:	CLEAR 1000 (1000 BYTES riservati a variabili alfanumeriche)

LASER 500:

Ortografia:	CLEAR nn,mm
Note:	nn e' un valore numerico che rappresenta il massimo valore della memoria destinata al BASIC, mm indica il numero di BYTES da riservare al BASIC, se non indicato assume valore 512.
Uso:	Imposta tutte le variabili numeriche a zero, e le variabili alfanumeriche a valore nullo, e definisce lo spazio di memoria da destinare alle variabili alfanumeriche.

CLOAD

Tipo:	COMANDO DI SISTEMA
Ortografia:	CLOAD"nome programma"
Note:	Il nome del programma puo' essere omesso.
Uso:	Carica nella memoria R.A.M. un programma leggendolo dal nastro del registratore a cassette.

Operazioni da eseguire per caricare un programma da nastro magnetico:

1)- Inserire nel registratore la cassetta che contiene il programma.

2)- Posizionare il nastro nel punto di inizio del programma.

3)- Digitare il comando CLOAD"nome programma",

4)- Premere il tasto PLAY del registratore.

5)- Quando appare la scritta READY sul video il programma e' stato caricato in memoria, ed occorre premere il tasto STOP del registratore.

6)- Se sul video compare la segnalazione LOADING ERROR modificare un poco la regolazione del volume del registratore, ripetendo le operazioni dall'inizio, fino a quando il programma viene caricato correttamente.

Esempio: Supponiamo di avere un nastro che contiene tre programmi che si chiamano PROG.1 PROG.2 PROG.3

Per caricare in memoria PROG.3 dopo aver posizionato il nastro all'inizio, seguire la procedura indicata sopra.

Sul video compariranno i seguenti messaggi:

```
CLOAD"PROG.3"
WAITING
FOUND T: PROG.1
FOUND T: PROG.2
LOADING T: PROG.3
READY
```

CLS

Tipo: ISTRUZIONE

Ortografia: CLS

Uso: Cancella tutto quello che compare sul video e posiziona il cursore in posizione HOME (angolo in alto a sinistra).

COLOR (LASER 210/310)

Tipo: FUNZIONE

Ortografia: COLOR I,J

Note: il parametro I e' un numero compreso tra 1 e 8
il parametro J puo' avere valore 0 o 1

Uso: Permette di selezionare i colori.
In MODE(0) (cioe' modo TESTO) i valori di I hanno la seguente codifica:

CODICE	COLORE
1	VERDE
2	GIALLO
3	BLU
4	ROSSO
5	MARRONE
6	CYANO
7	MAGENTA
8	ARANCIO

J determina il colore del bordo che puo' essere VERDE (0) oppure ARANCIO (1)

In MODE(1) (cioe' modo GRAFICO) I puo' assumere i valori 1 2 3 4 se J ha valore 0
In questo caso il bordo e' di colore VERDE.
Con i valori 5,6,7,8 J ha il valore 1 ed il bordo e' di colore marrone.

COLOR (LASER 500)

Tipo: FUNZIONE

Ortografia: COLOR I,J,K

Note: I,J,K indicano i colori del testo, del fondo e del bordo rispettivamente.
Se si omette uno dei prametri viene mantenuto il valore precedentemente scelto. Le virgole di separazione devono sempre essere scritte.

Uso: Permette di selezionare i colori per testo, fondo e bordo in relazione alla tavola dei colori.

I codici dei colori sono i seguenti:

0 NERO	4 ROSSO	8 GRIGIO 2	12 ROSSO CHIARO
1 BLU	5 MAGENTA	9 BLU CHIARO	13 MAGENTA CHIARO
2 VERDE	6 GIALLO	10 VERDE CHIARO	14 GIALLO CHIARO
3 CYANO	7 GRIGIO 1	11 CYANO CHIARO	15 BIANCO

Esempio: COLOR 0,4,6
COLOR ,,15

CONT

Tipo:	COMANDO DI SISTEMA
Ortografia:	CONT
Uso:	Fa proseguire l'esecuzione di un programma precedentemente interrotta con BREAK

COPY

Tipo:	ISTRUZIONE DI STAMPA
Ortografia:	COPY
Uso:	Permette di eseguire la copia del video su stampante. Non ha effetto per la pagina grafica, ma solo con il modo testo.

COS

Tipo:	FUNZIONE TRIGONOMETRICA
Ortografia:	COS(x)
Note:	x e' un valore numerico che indica l'angolo espresso in radianti.
Uso:	Da' il valore del coseno dell'angolo.
ESEMPIO:	10 A=COS(6.28) : PRINT A RUN .999995

CRUN

Tipo:	COMANDO DI SISTEMA
Ortografia:	CRUN "nome programma"
Note:	il nome del programma puo' essere omesso.
Uso:	Permette di caricare un programa dal nastro magnetico ed esegue il programma alla fine del caricamento. E' l'equivalente delle due istruzioni CLOAD e RUN (vedere uso di questi comandi)

CSAVE

Tipo: COMANDO DI SISTEMA

Ortografia: CSAVE"nome programma"

Note: Il nome del programma puo' essere omesso.

Uso: Permette di trasferire un programma dalla memoria del computer al nastro magnetico del registratore.

Operazioni da eseguire per effettuare il salvataggio del programma:

1)- Inserire una cassetta nel registratore

2)- Posizionare il nastro all'inizio, oppure in un punto prestabilito dove NON sia registrato nessun altro programma.

3)- Digitare il comando come indicato, e NON premere il tasto RETURN

4)- Premere contemporaneamente il tasto RECORD ed il tasto PLAY del registratore

5)- Premere il tasto RETURN

Dopo queste operazioni il cursore scompare dal video ed il salvataggio ha inizio.
Quando riappare il cursore il salvataggio e' terminato.

6)- Premere il tasto STOP del registratore.

7)- Dopo aver salvato un programma e' opportuno verificare che il salvataggio sia avvenuto in modo corretto con il comando VERIFY (vedere paragrafo relativo a tale comando).

CSGN (LASER 500)

Tipo: FUNZIONE DI CONVERSIONE

Ortografia: CSNG(x)

Note: x e' un argomento numerico

Uso: Converte x in un valore in singola precisione.

Esempio:

```
10 A# = 975.342145
20 PRINT A#;CSNG(A#)
RUN
975.342145  975.342
```

CVI/CVS/CVD (LASER 500)

Tipo: FUNZIONI DI CONVERSIONE

Ortografia:
CVI(x$)
CVS(x$)
CVD(x$)

Note: x$ e' un argomento alfanumerico.
Queste funzioni sono l'inverso delle funzioni MKI$ MKS$ e MKD$ (Vedere relativo paragrafo).

Uso: Conversione di un valore alfanumerico in un valore numerico.
CVI converte in un valore intero
CVS converte in un valore a precisione singola
CVD converte in un valore a precisione doppia

Esempio:

```
10 N$="TESTING STATEMENT"
20 PRINT CVI(N$),CVS(N$),CVD(N$)
RUN
17748     4.69115E-14     9.826534929158341D-30
```

DATA

Tipo: ISTRUZIONE

Ortografia: DATA [dato1],[dato2],[dato3], ... , ... , ...

Note: [dato1],[dato2],[dato3] etc. sono costanti numeriche, oppure costanti alfanumeriche.
Nel caso di costanti alfanumeriche che contenganc lo spazio o il segno virgola e' necessario scriverle tra virgolette, altrimenti le virgolette possono essere omesse.

Uso: Consente di conservare dei dati in memoria che possono essere letti con l'istruzione READ (vedere l'uso di questa istruzione).

Esempio:

```
10 DATA PROVA,1,2,"FINE DELLA PROVA"
20 READ A$,B,C%,D$
30 PRINT A$ : PRINT B : PRINT C% : PRINT D$
RUN
PROVA
1
2
FINE DELLA PROVA
```

DEF FN (LASER 500)

Tipo: ISTRUZIONE

Ortografia: DEF FN w(x,y)=[espressione]
DEF FN w(x)=[espressione]

Note: w e' il nome della funzione; x ed y sono i parametri usati dalla funzione. E' possibile indicarli entrambi, o indicare solo x.
[espressione] e' una serie di operazioni he la funzione svolge.

Uso: Definisce una funzione in grado di operare con i parametri x ed y. L'espressione che da' valore alla funzione e' una qualunque serie di operazioni scelte dal programmatore.

Esempio:

```
200 DEF FN A(N)= N-INT(N)
210 A=12.12345 : B=FN A(A)
220 PRINT A;B
RUN
12.12345  .123449
```

La funzione che e' stata definita con il nome A usa il parametro di nome N per determinare il valore dell'espressione N-INT(N) (in questo esempio l'espressione ricava la parte decimale di un numero troncandone la parte intera: l'opposto di INT).
Notare che si e' usato UN SOLO parametro anziche' due, inoltre e' soggetto ad una leggera imprecisione in quanto la parte decimale del numero 12.12345 e' .12345 e non .123449

DEF INT/SGN/DBL/STR (LASER 500)

Tipo: ISTRUZIONI

Ortografia: DEFtipo x-y,z-t, ..., ...

Uso: Definisce il tipo delle variabili usate nel programma, in relazione alla lettera dell'alfabeto con cui inizia il nome della variabile stessa, permettendo cosi' di non dovere indicare ogni volta i simboli $ % #

Il tipo puo' essere INT per intero, SGN per singola precisione,DBL per doppia precisione,oppure STR per stringa (alfanumeriche).

x y z e t sono le lettere iniziali dei nomi delle variabili che si definiscono.

Esempio: DEFINT A-E,K,Q-T le variabili il cui nome inizia per A,B,C,D,E,K,Q,R,S,T sono di tipo intero.

DEFSTR Z le variabili il cui nome inizia con Z sono alfanumeriche (stringhe).

Nel corso del programma e' possibile ridefinire una variabile usando i simboli # % $

DEF USR (LASER 500)

Tipo:	ISTRUZIONE
Ortografia:	DEF USRn=nn
Note:	n e' una cifra da 0 a 9 ed indica il numero della subroutine utente (USR) nn e' un valore numerico intero che da' l'indirizzo della subroutine in Assembly.
Uso:	Definisce l'indirizzo di partenza di una subroutine in Assembly definita dall'utente.
Esempio:	200 DEF USR0=24000 210 X=USR0(Y Z/2.89)

DELETE (LASER 500)

Tipo:	COMANDO DI SISTEMA
Ortografia:	DELETE [aa] - [bb]
Note:	[aa] e [bb] sono numeri di riga del programma.
Uso:	Cancella tutte le righe di programma comprese tra la riga [aa] e la riga [bb]. Se la riga bb non viene indicata cancella solo la riga [aa].

DIM

Tipo: ISTRUZIONE

Ortografia: DIM x(n1),a$(n2),b%(n3,n4), ... , ...

Note: x a$ b% stanno ad indicare vettori o tabelle di tipo diverso (numerici alfanumerici etc.) n1 n2 n3 n4 stanno ad indicare gli indici.

Uso: Permette di definire vettori e tabelle, per valori a piacere.
Puo' essere eseguita una sola volta nel corso del programma per la medesima variabile.
(vedere anche istruzione ERASE).

ESEMPIO: 10 DIM A%(25),A$(100,100)

DRAW (LASER 500)

Tipo: FUNZIONE GRAFICA

Ortografia: DRAW x,y

Note: x ed y sono argomenti numerici di tipo intero e possono assumere valore 0 oppure valori interi positivi con limite massimo definito dal tipo di grafica scelta (vedere l'uso delle funzioni GR e MOVE)

Uso: Disegna una linea a partire dalle ultime coordinate, fino alle nuove coordinate precisate con x ed y.

Esempio:

```
10 COLOR 15,0,0:GR1
20 MOVE (79,15)
30 DRAW (149,100):DRAW(9,100):DRAW(79,15)
40 GOTO 20
RUN
```

Sul video verra' disegnato un triangolo.
Per interrompere premere il tasto CTRL assieme al tasto C (BREAK).

END

Tipo: ISTRUZIONE

Ortografia: END

Uso: Definisce la fine logica di un programma.
Puo' essere inserita in qualunque punto del programma e ne interrompe l'esecuzione.

Esempio:

```
10 PRINT "PROVA" : END
20 PRINT "RIGA 20"
RUN                    (la riga 20 non verra' eseguita
PROVA                     poiche' si trova dopo END)
READY
```

ERASE (LASER 500)

Tipo: ISTRUZIONE

Ortografia: ERASE x,y,z, ... , ...

Note: x y e z sono nomi di vettori o tabelle

Uso: Cancella dalla memoria il vettore o la tabella indicata, permettendo il ridimensionamento (vedere istruzione DIM).

Esempio:

```
100 DIM A(20), B(20)
:
.
400 ERASE A,B
410 DIM A(12),B(12)
:
.
```

ERR/ERL (LASER 500)

Tipo: VARIABILI DI SISTEMA

Ortografia: ERR
ERL

Uso: Quando una subroutine di errore viene eseguita dopo una istruzione ON ERROR GOTO la variabile ERR contiene il codice dell'errore che si e' verificato, la variabile ERL invece contiene il numero di riga dove si e' verificato l'errore.
Se l'errore era causato da un comando dato in modo diretto, la variabile ERL contiene il valore 65535.

Esempio: Solitamente le variabili ERR ed ERL sono usate per mantenere il controllo del programma anche nel caso in cui si verifica un errore.
Questo e' possibile testando il valore di ERR o ERL (vedere codici errore sul manuale del computer).

ERROR

Tipo: ISTRUZIONE

Ortografia: ERROR x

Note: x e' un argomento numerico intero compreso tra 0 e 255 e corrisponde ad uno dei codici di errore (vedere tabella errori sul manuale).

Uso:
1)- Permette di usare gli errori codificati per segnalazioni (vedere TAVOLA ERRORI sul manuale del computer).

2)- Permette di codificare nuovi errori a proprio piacere, per fare segnalazioni.

Esempio 1:

```
10 S=10
20 T=5
30 ERROR S+T
40 END
RUN
STRING TO LONG IN LINE 30
```

Esempio 2:

```
110 ON ERROR GOTO 400
120 INPUT"QUANTO VUOI SCOMMETTERE";B
130 IF B>50000 THEN ERROR 210
 .
 .
400 IF ERR=210 THEN PRINT "TROPPO ALTA !"
410 IF ERL=130 THEN RESUME 120
 .
 .
```

EXP

Tipo: FUNZIONE MATEMATICA

Ortografia: EXP(x)

Uso: Da' il valore esponenziale di x, questo equivale ad elevare ad x il valore trascendentale e.
e = 2.17828
E' l'inverso della funzione LOG (vedere uso).

Esempio:

```
10 N=4 : T=EXP(N) : PRINT T
RUN
54.5982
```

FIX (LASER 500)

Tipo: FUNZIONE NUMERICA

Ortografia: FIX(x)

Note: x e' un argomento numerico

Uso: Tronca i decimali di un numero e da' la parte intera dello stesso.

Esempio:

```
200 A=10.998
210 PRINT FIX(A)
RUN
10
```

FOR ... TO ... STEP .... NEXT

Tipo: FRASE COMPOSTA DA PIU' ISTRUZIONI

Ortografia:

```
FOR x = n1 TO n2 STEP n3
  .
  .
  .
NEXT x
```

Note: x e' il nome di una variabile numerica
n1 e' il valore numerico di inizio del ciclo
n2 e' il valore numerico di fine ciclo
n3 e' il valore numerico di incremento

Uso: Tutte le istruzioni comprese tra la frase FOR e l'istruzione NEXT vengono ripetute fino a quando la variabile x (che ha valore iniziale n1) non assume il valore finale n2.
Ad ogni ripetizione l'incremento n3 viene sommato al valore della variabile x.
Se si intende un incremento pari ad 1 l'istruzione STEP n3 puo' essere omessa.
Il test sul valore di x avviene automaticamente dopo l'incremento causato dall'istruzione NEXT

Esempio:

```
10 FOR I = 0 TO 20 STEP 5
20 PRINT I;
30 NEXT I : PRINT
40 FOR I = 100 TO 10 STEP -10
50 PRINT I;
60 NEXT
70 END
RUN
  0   5  10  15  20
 100  90  80  70  60  50  40  30  20  10
```

FRE

Tipo: FUNZIONE DI SISTEMA

Ortografia: FRE(0)
FRE(x$)

Note: 0 ed x$ sono argomenti di comodo.

Uso: Da il numero di bytes liberi.
Le stringhe in BASIC possono avere differenti lunghezze, ma solitamente vengono manipolate nel corso del programma, per cui la memoria centrale non e' piu' ordinata.
Questa funzione permette di riorganizzare la memoria, recuperando spazio.

ESEMPIO:

```
PRINT FRE(0)
14542                 (o altro numero intero)
```

GOSUB

Tipo: ISTRUZIONE DI SALTO CON RITORNO

Ortografia: GOSUB nn

Note: nn e' un numero intero che fa riferimento ad una riga di programma esistente.

Uso: L'esecuzione del programma salta alla riga specificata e viene eseguito il sottoprogramma (subroutine). Quando viene eseguita l'istruzione RETURN l'esecuzione del programma ritorna alla istruzione successiva alla GOSUB che ha ordinato il salto.

Esempio:

```
10 PRINT "ESEGUO RIGA 10"
20 GOSUB 50 : PRINT "RITORNO DA SUBROUTINE"
30 PRINT "ED ORA ESEGUO RIGA 30"
40 END
45 REM
50 PRINT "HO ESEGUITO IL SALTO A RIGA 50"
60 PRINT "ED ORA PROSEGUO CON RIGA 60"
70 PRINT "RIGA 70 E RIGA 80"
80 RETURN
RUN
ESEGUO RIGA 10
HO ESEGUITO IL SALTO A RIGA 50
ED ORA PROSEGUO CON RIGA 60
RIGA 70 E RIGA 80
RITORNO DA SUBROUTINE
ED ORA ESEGUO RIGA 30
```

GOTO

Tipo: ISTRUZIONE DI SALTO INCONDIZIONATO DIRETTO

Ortografia: GOTO nn

Note: nn e' un numero intero, riferito ad una riga di programma.

Uso: Trasferisce l'esecuzione del programma alla riga indicata.

Esempio:

```
10 A =10 : GOTO 40
20 PRINT A*A
30 END
40 PRINT A
50 GOTO 20
RUN
 10
 100
```

GR (LASER 500)

Tipo: FUNZIONE GRAFICA

Ortografia: GRn

Note: n e' un numero compreso tra 0 o 5.
Se non e' indicato si sottintende 0

Uso: Definisce il tipo di pagina grafica.

GR0 da' 160 pixels orizzontali e 96 verticali, con 16 colori a disposizione (160H x 96V).

GR1 da' 160H x 192V con 16 colori, pero' ogni gruppo di 8 punti orizzontali, contando da sinistra a destra, puo' avere solo due dei 16 colori.Per vedere questo limite potete suddividere ogni riga orizzontale di 160 pixels in 20 celle, ognuna contenente 8 pixels.

GR2 da' 320H x 192V con possibilita' di usare 2 colori qualsiasi su tutto lo schermo, tra i sedici disponibili.

GR3 da' 160H x 192V con 16 colori a scelta per qualunque pixel.

GR4 da' 320H x 192V con disponibilita' di 16 colori. Come per GR1 ogni gruppo di 8 pixels orizzontali non puo' contenere piu' di 2 colori.

GR5 da' 640H x 192V con scelta di 2 colori tra i sedici a disposizione.

HEX$ (LASER 500)

Tipo: FUNZIONE DI CONVERSIONE

Ortografia: HEX$(x)

Note: x e' un argomento numerico intero.

Uso: Da' una stringa che contiene il valore esadecimale di x.

Esempio:

```
10 INPUT "NUMERO";N
20 A$=HEX$(N)
30 PRINT "DECIMALE=";N
40 PRINT "ESADECIMALE=";A$
50 END
RUN
NUMERO ? 42
DECIMALE=42
ESADECIMALE=2A
```

IF ... THEN ... ELSE

Tipo: FRASE COMPOSTA DA PIU' ISTRUZIONI

Ortografia:

```
IF n1 [confr] n2 THEN aa ELSE bb
IF n1 [confr] n2 THEN [istruz] : [istruz] ...
IF n1 [confr] n2 [op.log] n1 [confr] n2 then aa
```

Note: n1 ed n2 sono valori qualsiasi, ma dello stesso tipo (numerico o alfanumerico).

[confr] sta ad indicare un confronto eseguito per mezzo degli OPERATORI RELAZIONALI seguenti:

```
>     per indicare MAGGIORE
<      "     "     MINORE
=      "     "     UGUALE
<>     "     "     DIVERSO
>=     "     "     MAGGIORE O UGUALE
<=     "     "     MINORE o UGUALE
```

aa e' un numero intero che indica una riga di programma a cui saltare.

bb indica un' altra riga di programma

[istruz] sta ad indicare una qualsiasi istruzione BASIC

[op.log] indica un OPERATORE LOGICO (vedere relativo paragrafo).
Gli operatori logici di uso piu' comune sono: AND, OR, NOT.
Il LASER 500 dispone anche di XOR, IMP, EQV.

Uso: La frase IF... THEN ... ELSE permette la realizzazione di complessi confronti attraverso gli operatori RELAZIONALI, e numerose implicazioni attraverso gli operatori LOGICI.

In relazione a tali confronti ed a tali implicazioni si ha la possibilita' di eseguire un SALTO ad una riga di programma, per mezzo dell' istruzione THEN. Questo salto viene chiamato SALTO CONDIZIONATO DIRETTO.

E' possibile contrapporre altri confronti o implicazioni per mezzo del postulato ELSE, determinando un salto ad un'altra riga.

Non e' obbligatorio eseguire il salto ad una riga di programma. E' possibile inserire dopo l'istruzione THEN e/o dopo l'istruzione ELSE qualsiasi istruzione BASIC, determinando FRASI molto complesse, limitate solo dalle conoscenze individuali.

Esempio 1:

```
10 INPUT "NUMERO";N
20 IF N=10 THEN PRINT "FINE": END  ELSE PRINT N
30 GOTO 10
RUN
NUMERO ? 3
3
NUMERO ? 5
5
NUMERO ? 10
FINE
READY.
```

Esempio 2:

```
10 INPUT "PRIMO NUMERO";N1
20 INPUT "SECONDO NUMERO";N2
30 IF N1=1 AND N2=1 THEN  PRINT "FINE":END ELSE 500
500 IF N1>N2 THEN PRINT N1;"MAGGIORE DI";N2
510 IF N1<N2 THEN PRINT N1;"MINORE DI";N2
520 PRINT N1 "UGUALE A";N2
530 GOTO 10
RUN
PRIMO NUMERO ? 2
SECONDO NUMERO ? 8
2 MINORE DI 8
PRIMO NUMERO ?   10
SECONDO NUMERO ? 10
FINE
READY.
```

INKEY$

Tipo: FUNZIONE DI SISTEMA

Ortografia: INKEY$

Uso: Da' il carattere relativo all'ultimo tasto premuto. Il valore "" equivale a STRINGA NULLA.

Esempio:
```
10 PRINT "PREMI UN TASTO PER PROSEGUIRE"
20 A$ = INKEY$ : IF A$ = "" THEN 20
30 PRINT A$ : GOTO 10
```

INP

Tipo: FUNZIONE DI SISTEMA

Ortografia: INP (x)

Note: x e' un numero intero compreso tra 0 e 255

Uso: Da' il valore letto sulla porta di ingresso (INPUT PORT). Il valore di x e' limitato dalla Unita' Centrale del vostro computer (LASER 310, LASER 500 etc.).

Esempio:
```
100 A= INP(255): PRINT A
RUN  (comparira' un numero a caso tra 0 e 255)
```

INPUT

Tipo: ISTRUZIONE DI ASSEGNAZIONE

Ortografia: INPUT "messaggio"; x , x$, x%

Note: x x$ ed x% stanno ad indicare NOMI di variabili di qualsiasi tipo.

"messaggio" e' una COSTANTE ALFANUMERICA

Uso: Permette di introdurre dati dalla tastiera, assegnandoli alle variabili specificate dopo l'istruzione.
Se e' presente il "messaggio" esso viene scritto sul video, prima della richiesta dei dati.
Sul video compare un punto interrogativo che segnala di introdurre i dati da tastiera.

Esempio:
```
10 INPUT "INSERIRE UN NUMERO E UNA PAROLA";A,A$
20 INPUT "ORA INSERISCI TRE NUMERI";N1,N2,N3
30 PRINT "I DATI INSERITI SONO:" A;A$
40 PRINT N1;N2;N3
RUN
INSERIRE UN NUMERO E UNA PAROLA ? 125,PROVA
ORA INSERISCI TRE NUMERI ? 30,20,55
I DATI INSERITI SONO: 125 PROVA
30 20 55
```

INPUT#

Tipo: ISTRUZIONE DI ASSEGNAZIONE

Ortografia: INPUT# "nome file",[lista variabili]

Note: "nome file" e' il nome dell'archivio che deve essere letto dal nastro magnetico.

[lista variabili] e' una lista di nomi di variabili, a cui verranno assegnati i valori letti.

Uso: Trasferisce dal nastro magnetico una serie di dati letti in un FILE SEQUENZIALE.
Deve esserci assoluta concordanza tra i valori che erano stati precedentemente scritti con l'istruzione PRINT# (vedere questa istruzione), ed i valori che vengono letti ed assegnati alle variabili.

Esempio:

```
100 INPUT# "PROVA",A,B$,C,D
110 PRINT A,B$,C,D
RUN
FOUND D:PROVA
```

NOTA IMPORTANTE: IL FILE PROVA DEVE ESSERE GIA' STATO SCRITTO SUL NASTRO. PER QUANTO RIGUARDA LE OPERAZIONI CON IL REGISTRATORE VEDERE I COMANDI CSAVE E CLOAD

L' indicazione dell'archivio con la lettere D: sta ad indicare FILE di DATI.

INPUT$ (LASER 500)

Tipo: FUNZIONE ALFANUMERICA

Ortografia: INPUT$(x)

Note: x e' un argomento numerico.

Uso: Permette l'immissione di un numero prestabilito di caratteri, senza la comparsa del punto di domanda, e senza che i caratteri digitati compaiano sul video.

Esempio:

```
100 PRINT "PREMI  P PER PROSEGUIRE S PER STOP"
110 X$=INPUT$(1)
120 IF X$="P" THEN PRINT "PROSEGUO" : GOTO 500
130 IF X$="S" THEN PRINT "STOP" : STOP  ELSE 100
.
.
500 PRINT " IL PROGRAMMA E' PROSEGUITO": END
RUN
PREMI P PER PROSEGUIRE S PER STOP
PROSEGUO
IL PROGRAMMA E' PROSEGUITO
```

INSTR (LASER 500)

Tipo: FUNZIONE ALFANUMERICA

Ortografia: INSTR(I,x$,y$)

Note: I e' un argomento numerico compreso tra 1 e 255
x$ ed y$ sono valori alfanumerici (variabili o costanti alfanumeriche).

Uso: Da' la posizione della stringa y$ cercandola nella stringa x$ partendo dal punto I.
Se y$ compare piu' volte viene segnalata solo la prima volta.
Se I e' MAGGIORE della lunghezza di x$, o se x$ e' una stringa nulla, o se y$ non viene trovato la funzione INSTR ritorna il valore 0.
Se y$ e' una stringa nulla ("") la funzione ritorna il valore I oppure il valore 1.

Esempio:

```
10 X$="ABCDEB"
20 Y$="B"
30 PRINT INSTR(X$,Y$)
40 PRINT INSTR$(4,X$,Y$)
RUN
2
6
```

INT

Tipo: FUNZIONE NUMERICA

Ortografia: INT(x)

Note: x e' un argomento numerico.

Uso: Da' la parte intera dell'argomento x

Esempio:

```
10 A=INT(3.14) : PRINT A
RUN
3
```

JOY (LASER 500)

Tipo: FUNZIONE LETTURA JOYSTICK

Ortografia: JOY(x)

Note: x e' un numero compreso tra 0 e 5

Uso: JOY(0) ritorna un valore compreso tra 0 e 8 che rappresenta la posizione corrente del JOYSTICK di sinistra

JOY(1) ritorna un valore binario(0 o 1) relativo allo stato del pulsante di armamento (arm botton) del JOYSTICK di sinistra.
Il valore 1 indica pulsante premuto, mentre 0 indica pulsante rilasciato.

JOY(2) ritorna un valore binario (0 o 1) relativo allo stato del pulsante di sparo (fire botton) del JOYSTICK di sinistra.
Il valore 1 indica pulsante premuto, 0 pulsante rilasciato.

JOY(3) come JOY(0), ma per JOYSTICK di destra.

JOY(4) come JOY(1), ma per JOYSTICK di destra.

JOY(5) come JOY(3), ma per JOYSTICK di destra.

Esempio:

```
10 CLS
20 A=JOY(0):B=JOY(1):C=JOY(2)
30 D=JOY(3):E=JOY(4):F=JOY(5)
40 PRINT A,B,C,D,E,F
50 GOTO 20
RUN
```

(muovendo i due JOYSTICK e premendo i pulsanti potrete verificare i diversi valori che vengono trasmessi dal JOYSTICK)
Per interrompere: tasto CTRL + tasto C (BREAK)

KEY (LASER 500)

Tipo: FUNZIONE DI SISTEMA

Ortografia: KEY n ,"stringa"

Note: n e' un numero che identifica i TASTI FUNZIONE. I tasti FUNZIONE sono i tasti che si trovano sulla prima fila della tastiera del LASER 500.

I valori ammessi per n sono da 1 a 30.
Da 1 a 10 identificano i 10 tasti funzione.
Da 11 a 20 identificano i tasti funzione se premuti assieme al tasto SHIFT.
Da 21 a 30 identificano i tasti funzione se premuti con il tasto CTRL (CONTROL).

"stringa" e' una costante alfanumerica che racchiude uno o piu' comandi, che si vogliono attribuire al tasto funzione prescelto.

Uso: Permette di attribuire ai 30 tasti FUNZIONE dei comandi pre-programmati a nostra scelta.

Esempio: KEY3,"LIST"
attribuisce il comando LIST al tasto FUNZIONE 3

KEY18,"CLS:LIST"+CHR$(13)
attribuisce al tasto FUNZIONE 18 il comando CLS ed il comando LIST, inoltre il CHRS(13) che viene concatenato alla stringa con il segno + fa in modo che alla pressione del tasto F18 (cioe' SHIFT + TASTO F8) i due comandi CLS e LIST vengano eseguiti.
CHR$(13) e' il codice ASCII relativo al tasto di immissione RETURN.

Per avere l'elenco dei comandi pre-programmati sui tasti funzione si puo' usare il comando
KEY LIST
oppure se si vuole la stampa KEY LLIST

All'accensione del computer si hanno i seguenti comandi pre-programmati sui tasti F1-F10

F1 Cancella lo schermo e lista il programma
F2 Esegue il programma
F3 Seleziona modo testo 40 colonne
F4 Seleziona modo testo 80 colonne
F5 Sfondo NERO e testo BIANCO
F6 Sfondo AZZURRO e testo BLU
F7 Attiva scrittura standard
F8 Attiva scrittura INVERSE
F9 Spegne il BEEP della tastiera
F10 Accende il BEEP della tastiera

LEFT$

Tipo: FUNZIONE ALFANUMERICA

Ortografia: LEFT$(x$,I)

Note: x$ e' un valore alfanumerico
I e' un valore numerico

Uso: Permette di prelevare un certo numero di caratteri da una stringa, partendo dal primo carattere a sinistra, per la lunghezza I.

Esempio:

```
10 A$="ABCDEFGHI" : B$=LEFT$(A$,4) : PRINT B$
RUN
ABCD
```

LEN

Tipo: FUNZIONE ALFANUMERICA

Ortografia: LEN(x$)

Note: x$ e' un valore alfanumerico

Uso: Da' il numero di caratteri che costituiscono una stringa.

Esempio:

```
100 A$="ABCDEFG" : PRINT LEN(A$)
RUN
7
```

LET

Tipo: ISTRUZIONE DI ASSEGNAZIONE

Ortografia: LET variabile = espressione

Note: [variabile] e' il NOME di una variabile di qualunque TIPO (numerica etc.)

[espressione] puo' essere una costante, una variabile, una funzione oppure una, o piu', operazioni, o qualunque espressione composta.

Uso: Permette di ASSEGNARE alla [variabile] il valore della [espressione]
La parola LET e' facoltativa per i computers LASER, in quanto l' ASSEGNAZIONE viene effettuata con il solo segno = posto dopo il NOME di una variabile, sottointendendo l'istruzione LET.

Esempio:

```
30 LET A = 10 : LET B = A + A
40 C = 100 - B
50 PRINT A;B;C
RUN
10  20  80
```

LINE INPUT (LASER 500)

Tipo: ISTRUZIONE DI ASSEGNAZIONE

Ortografia: LINE INPUT x$

Note: x$ e' il nome di una variabile alfanumerica

Uso: Assegna alla variabile specificata la stringa alfanumerica che viene immessa da tastiera. Equivale all' istruzione INPUT (vedere), ma con LINE INPUT non compare il punto interrogativo sul video, e vengono accettati anche i segni di interpunzione (virgola etc.) che con l'istruzione INPUT causano la segnalazione EXTRA IGNORED.

Esempio:

```
100 LINE INPUT A$
110 PRINT A$
RUN
```

se digitate ABCDEFG,GHIJK (piu' tasto RETURN) sul video comparira' il testo inserito.

ABCDEFG,GHIJK

LIST

Tipo: COMANDO DI SISTEMA

Ortografia: LIST aa - bb

Note: aa e bb sono numeri di riga del programma.

Uso: Permette di avere sul video la lista del programma che si trova in memoria.

Le forme possibili sono le seguenti:

```
LIST          (lista tutto il programma)
LIST aa       (lista solo la riga aa)
LIST aa -     (lista da aa fino alla fine)
LIST - aa     (lista da inizio sino ad aa)
LIST aa - bb  (lista da aa sino a bb)
```

Per avere una PAUSA durante il listato premere la BARRA SPAZIO. Premerla un'altra volta per proseguire la lista.

LLIST

Tipo: COMANDO DI STAMPA

Ortografia: LLIST aa - bb

Note: Questo comando ha le stesse modalita' di esecuzione del comando LIST (vedere), solo che l'output avviene su stampante anziche' a video. Non ha effettto se la stampante non e' collegata al computer.

LOG

Tipo: FUNZIONE MATEMATICA

Ortografia: LOG(x)

Note: x e' un numero maggiore di 0

Uso: Da' il logaritmo di x in base e (e = 2.171828)
Per convertire in base 10 dividere per 2.3025

Esempio:

```
10 A=LOG(100)  : PRINT "LOG. BASE E =";A
20 PRINT "LOG. BASE 10 =";A/2.3025
RUN
LOG. BASE E = 4.60517019
LOG. BASE 10 = 2
```

LPOS (LASER 500)

Tipo: FUNZIONE DI STAMPA

Ortografia: LPOS(x)

Note: x e' un argomento numerico non significativo.

Uso: Da' la posizione della testina di stampa.
Puo' essere usata per definire i margini.

Esempio:

```
1000 IF LPOS(x)>60 THEN LPRINT CHR$(13)
```

LPRINT

Tipo: ISTRUZIONE DI STAMPA

Ortografia: LPRINT [lista di varabili o costanti]

Note: Non ha effetto se la stampante non e' collegata al computer. Verificare possibilita' di uso sul manuale della stampante.

Uso: Permette di stampare su carta la lista di variabili o costanti che sono indicate dopo il comando LPRINT.
Se non e' indicata niente provoca un salto di riga sulla stampante.

Esempio:

```
10 A$="ABCDE" :B=100
20 LPRINT "PROVA DI STAMPA";A$;A
RUN
```

MID$

Tipo: FUNZIONE ALFANUMERICA

Ortografia: MID$(x$,I,J)

Note: x$ e' un valore alfanumerico
I e J sono valori numerici

Uso: Permette di prelevare un certo numero di caratteri da una stringa, partendo dal carattere di posizione I , per una lunghezza pari a J.

Esempio:

```
10 A$="ABCDEFGHI" : B$=MID$(A$,3,4) : PRINT B$
RUN
CDEF
```

MKI$/MKS$/MKD$

Tipo: FUNZIONE DI CONVERSIONE

Ortografia: MKI$(x)
MKS$(x)
MKD$(x)

Note: x e' un valore numerico

Uso: Converte in una stringa di caratteri un valore numerico. (vedere anche CVI/CVS/CVD).

Esempio:

```
100 A$="LASER 500"
110 PRINT CVI(A$) : A%= CVI(A$)
120 PRINT CVS(A$) : B = CVS(A$)
130 PRINT CVD(A$) : C#= CVD(A$)
140 PRINT MKI(A%);MKS(B);MKD(C#)
RUN
16716                    (intero)
1.43152E-18              (singola precisione)
5.85250495516296D-25     (doppia precisione)
LA                       (stringa 2 bytes)
LASE                     (stringa 4 bytes)
LASER 50                 (stringa 8 bytes)
```

MOD (LASER 500)

Tipo: FUNZIONE MATEMATICA

Ortografia: x MOD y

Note: x ed y sono valori numerici.

Uso: Da' il resto della divisione di x con y.

Esempio: A = 10 MOD 3 (calcola 10/3 ed attribuisce il resto della divisione alla variabile A)

```
PRINT A
1
```

MODE (LASER 110/210/310)

Tipo: COMANDO GRAFICO

Ortografia: MODE(x)

Note: x puo' avere valore 0 oppure 1

Uso: Permette di selezionare il modo testo MODE(0) oppure il modo grafico MODE(1)

MON (LASER 500)

Tipo: COMANDO DI SISTEMA

Ortografia: MON

Uso: Attiva il MONITOR della memoria, cioe' il programma che permette di ispezionare, modificare, trasferire etc. i contenuti della memoria. Il MONITOR SYSTEM del LASER 500 permette inoltre la programmazione in linguaggio Assembly del microprocessore Z80.

E' necessario conoscere la notazione ESADECIMALE ed il linguaggio Assembly per poter usare con profitto questo importante programma.

Il manuale del computer ed altri testi specializzati trattano esaurientemente il linguaggio Assembly dello Z80.

I nuovi comandi attivati con il MONITOR sono:

: riempie o cambia il contenuto della memoria
M visualizza il contenuto di memoria
T trasferisce locazioni di memoria
I visualizza i valori di una porta di INPUT
O invia un valore ad una porta di OUTPUT
W salva il contenuto di memoria su nastro
R legge da nastro e carica in memoria
L disassembla e lista il contenuto di memoria
Z assembla i codici mnemonici in memoria
G esegue un programma in Codice Macchina
X visualizza i registri della CPU
P abilita/disabilita la stampante
\+ esegue somme esadecimali
\- esegue sottrazioni esadecimali
Q ritorna al BASIC

Esempio:

```
MON
<MON>
0008,0003+   (somma esadecimale)
000C
<MON>
```

MOVE (LASER 500)

Tipo: FUNZIONE GRAFICA

Ortografia: MOVE(x,y)

Note: x e y sono valori o espressioni numeriche comprese tra 0 ed il massimo valore ammesso dalla pagina grafica scelta (vedere GR).

Uso: Esegue il posizionamento nel punto di coordinate x ed y. Ha le stesse modalita' della funzione DRAW (vedere), ma non disegna nulla.
Serve solo per posizionarsi sul punto scelto.

Esempio:

```
100 GR 1 : MOVE(9,11) : DRAW(149,11)
110 DRAW(149,180) : DRAW(9,11)
120 MOVE(29,31) : DRAW(129,31)
130 DRAW(29,160):DRAW(29,31)
140 GOTO 140
RUN
```

per interrompere premere il tasto CTRL assieme al tasto C (BREAK).

NEW

Tipo: COMANDO DI SISTEMA

Ortografia: NEW

Uso: Cancella un programma dalla memoria.

NEXT

Tipo: ISTRUZIONE

Ortografia: NEXT x,y

Note: x ed y sono nomi di variabili numeriche che si riferiscono ad un ciclo FOR ... TO.
L'indicazione e' facoltativa per cicli semplici.
Per cicli concatenati il nome x si riferisce al ciclo piu' interno , mentre y si riferisce al ciclo esterno.

Uso: Dopo un ciclo FOR ... TO esegue l'incremento della variabile dipendente, ed esegue il test per ripetizione o fine ciclo.
Vedere istruzione FOR ... TO ... STEP ...

Esempio:

```
10 FOR I=1 TO 3 : FOR J=1 TO 3
20 PRINT I*J; : NEXT J,I
RUN
1 2 3 2 4 6 3 6 9
```

NULL (LASER 500)

Tipo: FUNZIONE DI EDITING

Ortografia: NULL x

Note: x e' un argomento numerico.

Uso: Fissa il numero di interlinee di stampa.

Esempio: NULL 2 dopo ogni PRINT stampa 2 caratteri nulli.

OCT$ (LASER 500)

Tipo: FUNZIONE DI CONVERSIONE

Ortografia: OCT$(x)

Note: x e' un argomento numerico.

Uso: Converte da base decimale a base ottale.

Esempio:

```
1000 PRINT OCT$(24)
RUN
30
```

ON ERROR GOTO (LASER 500)

Tipo: ISTRUZIONE DI SALTO

Ortografia: ON ERROR GOTO xx

Note: xx e' un numero di riga del programma.

Uso: Nel caso si verifichi un errore (in un qualsiasi punto del programma), salta alla riga indicata.

ON ... GOSUB

Tipo: ISTRUZIONE DI SALTO CONDIZIONATO CON RITORNO

Ortografia: ON k GOSUB aa, bb, cc, .. , ..

Note: k e' un valore o espressione numerica di cui viene considerato solo la parte intera.
aa bb cc sono numeri di riga di programma.

Uso: Come l'istruzione ON ... GOTO (vedere) solo che il salto e' fatto ad un sottoprogramma, con un ritorno alla riga che segue il comando ON .. GOSUB (vedere GOSUB e ON .. GOTO).

Esempio:

```
1000 ON K GOSUB 100 , , 300
1010 PRINT K
```

Se K=1 salta alla subroutine 100
Se K=3 salta alla subroutine 300
Per ogni altro valore prosegue con la riga 1010
Al ritorno dalle subroutine il programma prosegue lo stesso con riga 1010.

ON ... GOTO

Tipo: ISTRUZIONE DI SALTO CONDIZIONATO DIRETTO

Ortografia: ON k GOTO aa, bb,cc, .. , ..

Note: k e' un valore o espressione numerica di cui viene considerato solo la parte intera.
aa bb cc sono numeri di riga del programma.

Uso: Esegue il salto alla riga di programma indicata, in relazione al valore intero di k.
Se k=0 o se non e' precisato il numero di riga il programma prosegue alla riga seguente.

Esempio:
```
1000 ON K GOTO 100, 200, , 400
1010 PRINT K
```
Se K=1 salta a riga 100
Se K=2 salta a riga 200
Se K=3 prosegue con riga 1010
Se K=4 salta a riga 400

Per qualsiasi altro valore di K il programma prosegue con riga 1010.

OPTION BASE (LASER 500)

Tipo: COMANDO DI SISTEMA

Ortografia: OPTION BASE k

Note: k puo' avere valore 0 oppure 1

Uso: Stabilisce il valore minimo di INDICE per vettori e matrici.

Esempio: OPTION BASE 0 definisce 0 come valore minimo
OPTION BASE 1 definisce 1 come valore minimo.

OUT

Tipo: FUNZIONE DI SISTEMA

Ortografia: OUT i,j

Note: i e j sono valori tra 0 e 255

Uso: Trasmette un numero ad una porta di uscita del sistema (vedere anche funzione INP).
Occorre conoscere la struttura del sistema per usare con competenza questa funzione.

PEEK

Tipo: FUNZIONE DI SISTEMA

Ortografia: PEEK(x)

Note: x e' un valore intero che identifica una locazione di memoria (valore decimale).

Uso: Legge il valore contenuto nella locazione di memoria R.A.M. (compreso tra 0 e 255).
La possibilita' di lettura e' limitata dalla architettura del sistema, per cui e' necessario conoscere l'uso a cui sono destinate le diverse locazioni di memoria per poter usare con competenza la funzione PEEK.

Esempio:
```
100 PRINT PEEK(30779)
RUN
```
verra' scritto un numero compreso tra 0 e 255

POINT

Tipo: FUNZIONE GRAFICA

Ortografia: POINT(x,y)

Note: x e y sono numeri di valore minimo 0 e valore massimo dipendente dal tipo di pagina grafica.

Uso: Verifica se il punto di coordinate x e y e' acceso o se e' spento.
Se e' spento viene riportato il valore 0, altrimenti viene riportato il codice del colore.

Esempio:
```
80 SET(40,20)
90 IF POINT(40,20)<>0 THEN PRINT "ACCESO" .
RUN
ACCESO
```

POKE

Tipo: FUNZIONE DI SISTEMA

Ortografia: POKE x,y

Note: x identifica una locazione di memoria.
y e' un numero compreso tra 0 e 255.

Uso: Permette di inserire un valore (y) in una locazione R.A.M. non protetta.
Come la funzione PEEK (vedere) ha un uso molto particolare, per cui e' opportuno usarla solo se si conosce a fondo l'architettura del sistema.

Esempio:
```
10 POKE 30779,255 :PRINT PEEK(30779)
RUN
255
```

POS (LASER 500)

Tipo: FUNZIONE DI EDITING

Ortografia: POS(x)

Note: x e' un argomento numerico non significativo.

Uso: Da' la posizione in cui si trova il cursore. Il valore che indica la posizione piu' a sinistra e' 1.

Esempio:

```
300 IF POS(X) > 60 THEN PRINT CHR$(13)
```

PRINT

Tipo: ISTRUZIONE DI OUTPUT SU VIDEO

Ortografia: PRINT [lista]

Note: [lista] indica una serie di variabili, costanti etc. di cui si vuole avere la scrittura a video. Se non viene indicato nulla si ha un ritorno a capo del cursore.

Vedere anche PRINT@, PRINT USING, TAB e SPC

Uso: Permette di ottenere la scrittura su video degli elementi che si vogliono scrivere.

Esempio:

```
1000 A=350 : A$="PROVA"
1010 PRINT "SCRIVO SUL VIDEO";10;10*10+150;A;A$
RUN
SCRIVO SUL VIDEO 10 250 350 PROVA
```

PRINT@ (PRINT AT)

Tipo: ISTRUZIONE DI OUTPUT SU VIDEO

Ortografia: PRINT@x,[lista]

Note: [lista] e' una serie di valori che si vogliono scrivere sul video.
x e' la posizione ASSOLUTA del video in cui si vuole che inizi la scrittura. Il primo posto in alto a sinistra (HOME) ha il valore 0.

Uso: Scrive sul video quanto specificato.
Vedere istruzione PRINT

Esempio:

```
1000 PRINT@10,"PROVA"
RUN
```

la parola PROVA sara' scritta a partire dalla posizione 10 del video.

PRINT USING

Tipo: ISTRUZIONE DI OUTPUT SU VIDEO

Ortografia: PRINT USING [stringa];[lista]

Note: [stringa] e' una costante alfanumerica che contiene il formato di scrittura.
In particolare vengono usati i seguenti simboli:

! punto esclamativo
numero
+ piu'
- meno
. punto decimale
** doppio asterisco
$$ doppio dollaro
**$ doppio asterisco e dollaro
, virgola
% percento

ogni simbolo ha un preciso significato relativo al formato.

[lista] e' la serie di valori di cui si vuole l'output (vedere anche istruzione PRINT e derivate).

Uso: Permette di definire un formato di scrittura a video, per avere numeri allineati, o altre caratteristiche particolari.

Esempio:

```
PRINT USING "!";"ABCDEFG"
A
```

Il segno ! provoca la scrittura del solo primo carattere di una stringa.

```
PRINT USING "##.##";.78
0.78
```

Il segno # definisce il numero di cifre da rappresentare. Le cifre dopo il . sono le cifre decimali, quelle prima del . sono le cifre intere.

```
PRINT USING "+##.##";-68.95
-68.95
```

Il segno + definisce che il segno viene scritto prima del numero (situazione standard).

```
PRINT USING "##.##-";-68.95
68.95-
```

Il segno - mette il segno alla fine del numero.

PRINT USING "**#.#";-0.9
*-0.9
Il doppio asterisco fa precedere l'output da un asterisco che precede il numero.

PRINT USING "$$###.##";123.4
$ 123.40
Il doppio dollaro fa in modo che l'output del numero sia preceduto dal segno $

PRINT USING "**$###.##";456.78
*$456.78
Il segno **$ assomma l'effetto del segno ** e del segno $$

PRINT USING "#,###,###.##";1234567.123
1,234,567.12
Il segno , da' la rappresentazione del numero con le cifre a gruppi di tre.

Il segno % non e' usato per definire un formato, ma compare davanti ai numeri per cui e' stato sbagliato il formato. Ad esempio numero di 3 cifre intere il cui formato ne prevedeva solo 2.
PRINT USING "##.##";111.22
%111.22

PRINT#

Tipo: ISTRUZIONE DI OUTPUT

Ortografia: PRINT#"nome file",[lista]

Note: "nome file" indica il nome di un archivio sequenziale esistente su nastro magnetico.

[lista] e' una serie di valori da scrivere sul file sequenziale.

Uso: Trasferisce i dati sul nastro magnetico.

Esempio: 100 PRINT#"prova",A$,B$,C$

I valori contenuti nelle variabili A$ B$ e C$ vengono scritti in un file su nastro magnetico.
Non ammessi valori NULLI di variabili stringa.
Per la lettura dei dati vedere istruzione INPUT#

Vedere anche comandi CSAVE e CLOAD.

RANDOMIZE (LASER 500)

Tipo: FUNZIONE DI SISTEMA

Ortografia RANDOMIZE

Uso: Fa iniziare una nuova sequenza di numeri pseudo-casuali generati dalla funzione RND.
Vedere uso della funzione RND

READ

Tipo: ISTRUZIONE DI ASSEGNAZIONE

Ortografia: READ [lista variabili]

Uso: Legge i dati memorizzati nell'area BASIC (vedere istruzione DATA), ed ASSEGNA i valori letti alle variabili i cui NOMI sono specificati nella [lista variabili] separati da virgole.

Occorre che il TIPO delle variabili precisate concordi con il valore indicato nell'istruzione DATA. Inoltre, il numero di valori letti, deve essere uguale ai valori esistenti in memoria.

Esempio:

```
100 DATA 100,PROVA,355.25
110 READ A%,A$,A  :PRINT A%;A$;A
RUN
100  PROVA  355.25
```

REM

Tipo: ISTRUZIONE DI NON OPERAZIONE (NOP)

Ortografia: REM [commento]

Uso: Non viene eseguita alcuna operazione. Il commento posto dopo l'istruzione REM documenta il programma e permette, a chiunque legga il listato, di comprendere meglio i diversi passaggi e le operazioni che vengono eseguite.

E' buona regola di programmazione documentare nel modo piu' chiaro possibile qualunque programma, anche se breve.

Esempio:

```
100 REM QUESTA E'UNA RIGA DI COMMENTI
110 REM
120 REM INIZIO DEL PROGRAMMA
```

RENUM

Tipo: COMANDO DI SISTEMA

Ortografia: RENUM nn,vn,in

Note:
nn = nuovo numero di riga
vn = vecchio numero di riga
in = incremento

Uso: Rinumera tutto o in parte il programma.

Esempio: RENUM rinumera tutto. La nuova lista iniziera' con riga 10 e proseguira' di 10 in 10.

RENUM 300,50 rinumera tutto. La nuova lista iniziera' con riga 300 incrementando di 50 in 50

RES (LASER 500)

Tipo: FUNZIONE GRAFICA

Ortografia: RES (x,y)

Note: x ed y sono valori numerici interi positivi, compresi tra 0 ed il valore massimo previsto dalla pagina grafica prescelta (vedere funzione GR).

Uso: Spegne il punto di coordinate x e y nella pagina grafica prescelta con la funzione GR.
E' l'opposto della funzione SET (x,y) (vedere).

Esempio:

```
100  GR1  :  FOR I = 0 TO 50 : J=I :  SET (I,J)
110  NEXT I : FOR T = 1 TO 500 : NEXT T
120  FOR I = 0 TO 50 : RES (I,J) : NEXT I
RUN
```

(disegna una riga diagonale e poi la cancella.)

RESET (LASER 110/210/310)

Tipo: FUNZIONE GRAFICA

Ortografia: RESET (x,y)

Note: x ed y indicano le coordinate dei punti (pixels) della pagina grafica.

Uso: Spegne il punto di coordinate x ed y.
E' la funzione opposta a SET (x,y) (vedere).

Esempio:

```
100 MODE(1)
110 FOR X=1 TO 30 : SET (X,X+1) : NEXT
120 SOUND 20,5
130 FOR X=1 TO 30 : RESET(X,X+1) : NEXT
```

RESTORE

Tipo: ISTRUZIONE

Ortografia: RESTORE

Uso: Ripristina il puntatore al PRIMO valore della PRIMA istruzione DATA (vedere istruzione DATA ed istruzione READ).

Esempio:

```
10 FOR I = 1 TO 3  : READ A,B,C
20 RESTORE : READ E,F,G
30 PRINT A;B;C;D;E;F;G
40 DATA 10,20,30
RUN
10  20  30  10  20  30
```

RESUME (LASER 500)

Tipo: FUNZIONE DI SISTEMA

Ortografia: RESUME

Uso: Riprende l'esecuzione di un programma dopo che si e' interrotto a causa di un errore.

Esempio: RESUME (o RESUME 0) il programma riprende dalla riga in cui si e' interrotto.

RESUME NEXT il programma riprende alla linea immediatamente successiva a quella in cui si e' verificato l'errore.

RESUME [numero riga] il programma riprende dal numero di riga specificato.

RETURN

Tipo: ISTRUZIONE

Ortografia: RETURN

Note: NON CONFONDERE l'istruzione RETURN con il TASTO che ha lo stesso nome.

Uso: Indica la fine di un sottoprogramma (subroutine). Vedere GOSUB e ON GOSUB .

Esempio:

```
100 GOSUB 1000
110 PRINT "FINE"
120 END
1000 PRINT "SUBROUTINE" : RETURN
RUN
SUBROUTINE
FINE
```

RIGHT$

Tipo: FUNZIONE ALFANUMERICA

Ortografia: RIGHT$(x$,i)

Note: x$ e' una costante o una variabile alfanumerica. i e' una costante o variabile numerica.

Uso: Preleva dalla stringa x$ un certo numero di caratteri, precisato da i, a partire dall'ultimo carattere a destra della stringa.

Vedere anche MID$ e LEFT$

Esempio:

```
1000 Z$="ABCDEF" : PRINT RIGHT$(Z$,4)
RUN
CDEF
```

RND

Tipo: FUNZIONE NUMERICA

Ortografia: RND(k)

Note: k e' un valore numerico.

Uso: Genera un numero PSEUDO-CASUALE.

Esempio: Per LAER 110/210/310:

l'argomento (k) deve essere SEMPRE >= 0

RND(K) se K > 0 genera un numero tra 1 e K
RND(K) se K = 0 genera un numero tra 0 e 1

Per LASER 500:

Vedere istruzione RANDOMIZE.

RND(K) genera sempre un numero tra 0 e 1 qualunque sia il valore di K.

Se K = 0 ripete l'ultimo numero generato.
Se K > 0 genera un nuovo numero della sequenza.
Se K < 0 o non indicato ripete la sequenza.

RUN

Tipo: COMANDO DI SISTEMA

Ortografia: RUN nn

Note: nn indica un numero di riga.

Uso: Esegue il programma BASIC che c'e' in memoria, a partire dalla riga indicata. Se non viene indicato il numero di riga il programma parte dalla prima riga.

SET

Tipo: FUNZIONE GRAFICA

Ortografia: SET(x,y)

Note: x e y indicano le coordinate dei punti della pagina grafica prescelta.

Vedere istruzioni MODE, GR, RESET, RES, DRAW, MOVE, COLOR (in relazione al computer usato).

Uso: Accende il punto (pixel) di coordinate x e y.

Esempio:

```
100  MODE  (1) : FOR I = 1  TO  30 :  SET(I,I+1)
110  NEXT I : SOUND 25,8
```

(per LASER 500 sostituire MODE(1) con GR 1)

SGN (LASER 500)

Tipo: FUNZIONE NUMERICA

Ortografia: SGN(x)

Note: x e' un valore od un espressione numerica.

Uso: Restituisce un numero che indica il segno dell'argomento tra parentesi.
Se (x) = 0 la funzione riporta 0
Se (x) > 0 la funzione riporta 1
Se (x) < 0 la funzione riporta -1

Esempio:

```
100 N =10 : B=-5 : A= 0
110 PRINT SGN(N);SGN(B);SGN(A)
RUN
1 -1  0
```

SIN

Tipo: FUNZIONE TRIGONOMETRICA

Ortografia: SIN(x)

Note: x e' un numero che esprime il valore di un angolo in RADIANTI. (1 RADIANTE = 57.296 GRADI)

Uso: Da' il valore del seno dell'angolo indicato.

Esempio:

```
100 A = 3.141592654
110 PRINT SIN(A/4)
RUN
0.707106781
```

SOUND

Tipo: FUNZIONE SONORA

Ortografia: SOUND n,d

Note: n e' un numero tra 0 e 31 ed indica una nota secondo la tabella seguente:

n			n		
0	Pausa		16	C4	Do
1	A2	La	17	C#4	Do#
2	A#2	La#	18	D4	Re
3	B2	Si	19	D#4	Re#
4	C3	Do	20	E4	Mi
5	C#2	Do#	21	F4	Fa
6	D3	Re	22	F#4	Fa#
7	D#3	Re#	23	G4	Sol
8	E3	Mi	24	G#4	Sol#
9	F3	Fa	25	A4	La
10	F#3	Fa#	26	A#4	La#
11	G3	Sol	27	B4	Si
12	G#3	Sol#	28	C5	Do
13	A3	La	29	C#5	Do#
14	A#3	La#	30	D5	Re
15	B3	Si	31	D#5	Re#

d e' un numero tra 1 e 9 ed indica la durata della nota secondo la seguente tabella:

d		d		d	
1	1/8	4	1/2	7	3/2
2	1/4	5	3/4	8	8/4
3	3/8	6	4/4	9	12/4

Uso: Attiva il generatore di suoni, che emettera' la nota scelta per la durata richiesta.

Esempio: SOUND 11,8

SPACE$ (LASER 500)

Tipo: FUNZIONE ALFANUMERICA

Ortografia: SPACE$(x)

Note: x e' un numero o una espressione numerica.

Uso: Da' una stringa che contiene il numero di spazi CHR$(32) precisati con x.

Esempio:

```
10  N$ = SPACE$(10) : PRINT "**";N$;"*****"
RUN
**          *****
```

SPC (LASER 500)

Tipo: FUNZIONE DI EDITING

Ortografia: SPC(x)

Note: x e' un valore o espressione numerica. Questa funzione deve essere usata con l'istruzione PRINT o LPRINT . Vedere queste istruzioni e le loro derivate (PRINT@, PRINT USING etc.).

Uso: Permette di separare con spazi vuoti cio' che si vuole scrivere.

Esempio:

```
1000  Z$="ABCDEF" : PRINT Z$SPC(8)Z$SPC(2)Z$
RUN
ABCDEF        ABCDEF  ABCDEF
```

SQR

Tipo: FUNZIONE MATEMATICA

Ortografia: SQR(x)

Note: x e' un argomento o un'espressione numerica.

Uso: Da' la radice quadrata del valore tra parentesi.

Esempio:

```
PRINT SQR(100);SQR(81);SQR(9)
10  9  3
```

STEP

Tipo:	ISTRUZIONE
Ortografia:	STEP k
Note:	k e' un valore numerico.
Uso:	Puo' essere usata solo assieme all'istruzione FOR ... = ... TO ... STEP ... per l'uso vedere tale istruzione.

STOP

Tipo:	ISTRUZIONE DI ARRESTO
Ortografia:	STOP
Uso:	Ferma l'esecuzione del programma alla riga in cui si trova l'istruzione STOP. Vedere anche istruzione CONT.

STR$

Tipo:	FUNZIONE DI CONVERSIONE
Ortografia:	STR$(x)
Note:	x e' un valore o espressione numerica. Vedere anche funzione VAL.
Uso:	Converte un numero in una stringa.

Esempio:

```
100  A = 1255 : B = 125.378 : C$=STR$(B)
110  PRINT STR$(A);C$
RUN
1255  125.378
```

SWAP (LASER 500)

Tipo: ISTRUZIONE DI ASSEGNAZIONE

Ortografia: SWAP [variabile 1] , [variabile 2]

Uso: Scambia il valore contenuto in [variabile 1] con il valore di [variabile 2].

Esempio:
```
100 A=100 : B=200
110 PRINT "PRIMA DI SWAP A=";A;"B=";B
120 SWAP A,B : PRINT "DOPO SWAP A=";A;"B=";B
RUN
PRIMA DI SWAP A=100 B=200
DOPO SWAP A=200 B=100
```

TAB

Tipo: FUNZIONE DI EDITING

Ortografia: TAB(x)

Note: x e' un valore o espressione numerica.

Uso: Usata con l'istruzione PRINT e sue derivate (LPRINT,PRINT@ etc.) permette di tabulare in un punto prestabilito.

Esempio:
```
100 PRINT "+++++"TAB(10)"*****"
RUN
+++++     *****
```

TAN

Tipo: FUNZIONE TRIGONOMETRICA

Ortografia: TAN(x)

Note: x indica un angolo in RADIANTI.
(1 RADIANTE = 57.296 GRADI SESSAGESIMALI)

Uso: Da' la tangente dell'angolo

Esempio:
```
PRINT TAN(3.141592654/2)
2
```

TEXT (LASER 500)

Tipo: FUNZIONE

Ortografia: TEXT x

Note: x puo' avere valore 40 oppure 80.

Uso: Attiva il modo testo precisato con x.

Esempio:
TEXT 40 abilita MODO TESTO 40 colonne x 25 righe
TEXT 80 abilita MODO TESTO 80 colonne x 25 righe

TROFF (LASER 500)

Tipo: COMANDO DI SISTEMA

Ortografia: TROFF

Uso: Disabilita il comando TRON (TRACE ON) che permette il debugging del programma.
Vedere uso del comando TRON.

TRON (LASER 500)

Tipo: COMANDO DI SISTEMA

Ortografia: TRON

Uso: Attiva il TRACE ON, cioe' la traccia dell'esecuzione del programma per il debugging.
Con TRON e' possibile verificare l'esecuzione del programma.
Dopo aver dato il comando TRON occorre far eseguire il programma con il comando RUN.
Sul video compariranno, tra parentesi quadre, i numeri delle righe di programma, nell' ordine esatto in cui vengono eseguite.
Ogni istruzione viene eseguita correttamente, compreso le istruzioni di OUTPUT (PRINT, LPRINT etc.).

Esempio:

```
10 K=10
20 FOR J=1 TO 2
30 L=K+10
40 PRINT J;K;L
50 K=K+10
60 NEXT J
70 END
TRON
READY
RUN
[10] [20] [30] [40]  1  10  20
[50] [60] [30] [40]  2  20  30
[50] [60] [70]
READY
TROFF
READY
```

USR

Tipo: FUNZIONE DI SISTEMA

Ortografia: USR(x)

Note: x e' un valore numerico.

Uso: Esegue un salto con ritorno ad una subroutine in linguaggio Assembly definita dall'utente.
Vedere anche istruzione DEF USR (LASER 500).

VAL

Tipo: FUNZIONE DI CONVERSIONE

Ortografia: VAL(x$)

Note: x$ e' una costante o variabile alfanumerica.

Uso: Converte le cifre contenute in una stringa in numeri. Se la stringa non contiene numeri, la funzione restituisce il valore 0.
Se la stringa contiene lettere e numeri, vengono convertiti solo i numeri piu' a sinistra, che non siano preceduti da lettere dell'alfabeto o altri simboli.
L'unico simbolo ammesso e' il segno - (meno) che indica numeri negativi.

Esempio:

```
10 A$="123.25" : B$="SZ-12vx" : C$=" -12.02FG31"
20 A=VAL(A$) : B=VAL(B$) : C=VAL(C$)
30 D=VAL(C$)*2
40 PRINT A;B;C;D
RUN
123.25  0  -12.02  -24.04
```

VARPTR (LASER 500)

Tipo: FUNZIONE DI SISTEMA

Ortografia: VARPTR(x)

Note: x indica il NOME di una variabile di QUALSIASI TIPO.

Uso: Da' l'indirizzo della LOCAZIONE DI MEMORIA R.A.M in cui e' allocata la variabile di cui si indica il NOME.
Viene usata solitamente per inviare questo indirizzo ad una subroutine in Assembly.

Esempio:

```
100  X=USR(VARPTR(Y))
```

WAIT (LASER 500)

Tipo: ISTRUZIONE DI ARRESTO (PAUSA)

Ortografia: WAIT [N.porta],i,j

Note: [N.porta] indica il numero di porta di I/O
i e j sono valori tra 0 e 255
se j non e' indicato si sottointende valore 0.

Uso: Ferma l'esecuzione del programma fino a quando la porta indicata non ha un valore preciso.
Il valore letto sulla porta viene confrontato con un XOR (OR ESCLUSIVO) con il valore j.
Il risultato di XOR viene poi sottoposto ad un AND con il valore indicato con i.
Se il risultato finale e' 0 viene letto un altro valore ed il programma resta in attesa, altrimenti prosegue con l'istruzione successiva.

Esempio: WAIT 32,2

WHILE ... WEND (LASER 500)

Tipo: ISTRUZIONE COMPOSTA

Ortografia:

```
WHILE [confronto]
[istruzioni]
      .
      .
[istruzioni]
      .
      .
WEND
```

Note: [confronto] indica uno o piu' confronti eseguiti con operatori logici e relazionali.
[istruzioni] indica una o piu' istruzioni disposte in una o piu' righe di programma.

Uso: Consente di ripetere una sequenza di istruzioni, o righe di programma, fino a quando non diventa VERO il [confronto] previsto.
Ha analogia con il ciclo FOR ... NEXT ma presenta minori inconvenienti d'uso.

Esempio:

```
10 I=0
20 WHILE A<26
30 I=I+1
40 A=I*I
50 PRINT I;A
60 WEND
70 PRINT "FINE CICLO"
RUN
1   1
2   4
3   9
4  16
5  25
FINE CICLO
```

WIDTH (LASER 500)

Tipo: FUNZIONE DI EDITING

Ortografia:
```
WIDTH n
WIDTH LPRINT n
```

Note: n e' un valore o espressione numerica, compreso tra 15 e 255 (valore normale 72).

Uso: Definisce la lunghezza della riga di output.

Esempio:
```
100 A$="ABCDEFGHIJKLMNOPQRST":PRINT A$
RUN
ABCDEFGHIJKLMNOPQRST
WIDTH 16
READY
RUN
ABCDEFGHIJKLMNOP
```

WRITE (LASER 500)

Tipo: ISTRUZIONE DI OUTPUT

Ortografia:
```
WRITE [lista],[lista], ... , ...
```

Note: [lista] rappresenta una serie di valori che si vogliono scrivere (costanti, variabili etc.)

Uso: Come l'istruzione PRINT e derivate (vedere), provoca l'uscita sul video dei valori indicati.

Le costanti e le variabili alfanumeriche vengono scritte sul video tra virgolette. Se si indicano piu' valori vengono trascritti separati da virgole.

Esempio:
```
10 A=80 : B=90 : C$="PROVA"
20 WRITE A,B,C$
RUN
80 , 90 , "PROVA"
```

---

* scuola scheidegger *

Corso di informatica
(I livello)

*   TEST   *

NOME . . . .

COGNOME . . . .

---

* scuola scheidegger *

Corso di informatica (I livello)

L.07/T f91.1

Leggete attentamente le seguenti righe di programma:

```
10 LET A=5 : LET B=100 : LET C=35
20 LET D=A : LET A=B : LET B=C : LET C=D
30 LET D=D+1
40 END
```

Scrivete il CONTENUTO delle variabili A,B,C e D DOPO l'esecuzione del programma di cui sopra:

Variabile A = . . .

Variabile B = . . .

Variabile C = . . .

Variabile D = . . .

Leggete attentamente il programma che segue:

```
 10 CLS
 20 DIM A(3,3)
 30 FOR I=3 TO 1 STEP -1
 40 FOR J=3 TO 1 STEP -1
 50 READ A(I,J) : NEXT J,I
 60 DATA 9,6,3,6,4,2,3,2,1
 70 FOR I=1 TO 3
 80 FOR J=1 TO 3
 90 PRINT A(I,J);
100 NEXT J
110 PRINT
120 NEXT I
130 END
```

DOPO l'esecuzione del programma di cui sopra sul video compariranno dei numeri disposti su TRE RIGHE (ogni riga con TRE numeri).

Scrivete TALI numeri nello STESSO modo in cui li scrivera' il computer:

....

....

....

Scrivete l'istruzione che permette di conservare NEL PROGRAMMA dei DATI:

100 ....

Quanti sono gli ELEMENTI della variabile dimensionata con DIM N(2,3,4) ?

RISPOSTA: La variabile ha ...... elementi.

---

* scuola scheidegger *

Corso di informatica (I livello) L.08/T f91.1

1)- Leggete attentamente le seguenti righe di programma:

```
10 LET A=5 : LET B=12.25 : LET C=2354
20 PRINT#''NUMERI'',A;CHR$(13);B;CHR$(13);C;CHR$(13);
30 END
```

Quale operazione viene eseguita dal breve programma di cui sopra ?

. . . .

2)- Leggete attentamente il programma che segue:

```
10 CLS
20 DIM A(3)
30 FOR I=1 TO 3
40 INPUT#,A(I)
50 NEXT I
60 END
```

DOPO l'esecuzione del programma di cui sopra (e supponendo di aver eseguito il programma di cui al punto 1), quali valori conterra' la variabile indicizzata A ?

RISPOSTA : A(1)= . . . A(2)= . . . A(3)= . . .

3)- Scrivete l'istruzione che permette di SCRIVERE in un FILE SEQUENZIALE il CONTENUTO di DUE VARIABILI ALFANUMERICHE:

100 ....